REMARKS

Applicants acknowledge that the Office Action dated September 15, 2004 has been made Final. Accordingly, the foregoing Amendment cancels claims 5, 9 and 19, and rewrites claims 6 and 8 in independent form. In addition, minor amendments have been made to correct certain formal matters. Reconsideration and withdrawal of the final rejection is hereby requested, for the reasons set forth hereinafter.

Applicants note that the copy of the final Office Action which they have received does not include a signed copy of PTO-1449, indicating that the Information Disclosure Statement filed August 9, 1999 has been considered, including in particular, Japanese Patent document JP 6-332807. As noted in the remarks which accompanied the Amendment submitted June 21, 2004, a "concise explanation of the relevance" of the latter Japanese patent document was included in the specification of the present application. Moreover, an English language abstract of the latter document was also submitted. Accordingly, Applicants respectfully submit that the Information Disclosure Statement filed August 9, 1999 is proper and confirmation is requested that the documents cited therein have been considered of record in this application.

Claims 5, 6, 8-16 and 18-22 have been rejected under 35 U.S.C. §102(e) as anticipated by Green et al. (U.S. Patent No. 6,111,888). Nevertheless, as set forth in greater detail below, Applicants respectfully submit that all such claims distinguish over Green et al., whether considered by itself, or in combination with other references.

The present invention is directed to a distributed control system, in which a plurality of built-in systems are connected by a field network, such as a CAN (Controller Area Network), a foundation field bus or the like. Such systems have been utilized, for example, in power train control systems for vehicles, field instrument control systems for factory automation or process automation and control systems for medical instruments or a robot. One object of the present invention is to provide such a system, which incorporates means for assuring real-time operability of application programs that are running on the system. To this end, an application program is divided into two parts: (i) a main body of the application program, and (ii) "module start control means" which executes communication and other processing.

An important feature of the present invention, therefore, is that, in order to realize real-time communication, within the respective network nodes, the module start control means is provided separately from the real time communication processing control means. By contrast, a central feature of the Green et al. patent is that, in order to realize real-time communication by making use of a standard protocol CAN, one node attached to a data bus is selected as the master node, which periodically transmits synchronization signals, such that time "TDMA-like" communication can be achieved, using the CAN protocol. (See, for example, col. 4, lines 1-5.)

Claims 6 and 14 of the present application recite in particular that the real-time communication processing control means, which controls execution of message objects based on assigned priorities, referring to the message object configuration means, is a software module executed in a task. In Green et al., on the other hand, a hardware CAN controller is expressly indicated for use as the real-time communication processing control means. Thus, for example, at col. 3, lines 38-40, Green et al. states that, "The present invention provides a method and apparatus by which standard CAN processors and related hardware are used in a real-time, deterministic processing system." Consistently, at col. 5, lines 63-65, Green et al. indicates that, "CAN controller 106 is physically connected to a bus 104 through CAN transceiver 105." Nowhere does Green et al. contemplate or suggest the provision of a real-time communication processing means in the form of a software module which executes the function as a task. Accordingly, claims 6 and 14 distinguish over Green et al.

Claim 18 further specifies that a message object configuration information storing means includes information indicating whether a message object is an in-unit communication or an inter-unit communication. (Claim 8 is similarly limited.) Figures 2-4 in Green et al., however, do not represent data stored in the memory means. Rather, they show timing diagrams for the CAN bus, and the manner of data flow on the CAN bus. Accordingly, the data are limited to messages communicated between nodes, and contain no such indication regarding whether the respective message objects are for in-node communication or for inter-node communication. That is, Figs. 2-4 simply show the current instantaneous message transmission order, and do not indicate the priority of the respective messages.

With regard to independent claim 10, Applicants note that, in the present invention, each of the respective nodes is provided with a module configuration information storing means, in which an execution order is stored for all application programs on the node in question. The master node in Green et al., on the other hand, simply sends synchronous messages. Nowhere does Green et al. indicate that the master node stores an execution order for the application programs. Moreover, the priority, as described in Green et al., is an inter-node priority in the CAN protocol, which is fundamentally different from the inter-message priority within a node, according to the application program in the present invention.

The module start control means in the present invention is always operating, and controls the start of respective application program modules. The software router in Green et al., however, is operated only at the time when a message arrives from the CAN bus and starts an application program in response to the message which has arrived. Accordingly, the module start control means in the present invention and the software router in Green et al. are fundamentally different.

As mentioned previously with regard to claims 6 and 14, the real-time communication processing control means according to the present invention, and the CAN controller in Green et al. are very different. The real-time communication processing control means in the present invention controls message transmission and receipt, based on the execution order of the application program modules, and thus performs a control processing superior to the protocol processing according to Green et al. Furthermore, as has also been noted previously, the priority in messages in the present invention is an inter-message priority within a node, which differs from the inter-node priority in the CAN protocol in Green et al.

With regard to claim 11, the control module means according to the present invention controls the start of the application programs, but does not control communication processing. The arguments set forth in Office Action with regard to claim 11 seem not to be directed to that claim, but rather claim 14. In this case, the real-time communication processing control means according to the present invention is, as explained previously, for controlling the start of transmission and receipt processing of all message objects in a node, and not for controlling the sending of synchronism messages. Finally, Green et al. nowhere discloses that the real-time communication processing control means is executed in a task.

Claims 12 recites that the module start control means of claim 10 is a function included in an operating system. As noted previously with regard to claim 10, the module start control means in the present invention is operating continuously, and controls the start of respective application program modules.

On the other hand, the software router in Green et al. is operating only when a message has arrived from the CAN bus, and starts an application program in response to the message arrival. Thus, the module start control means as defined in the claims and the software router in Green et al. are not comparable. Furthermore, Green et al. indicates at col. 11, lines 57 and 58, that the "software router 516 is an application executed by CPU 507." The structures in application software and operating system software are also not comparable.

Claims 13 recites that the program module configuration information storing means according to claim 10 includes a designation of a software module which is to be executed next in sequence. The software router in Green et al., on the other hand, grasps an application program to be started in response to the arrival of a message, but does not store an execution order for all the application programs contained within a particular node. Accordingly, claim 13 distinguishes over Green et al. for this additional reason as well. (Claim 20 is similarly limited, and thus distinguishes for the same reasons.)

With regard to claim 15, Applicants note that, as has been indicated previously regarding claim 6 and 14, in Green et al., a hardware CAN controller and receiver are provided as the real-time communication processing control means. Nowhere does Green et al. indicate that the real-time communication processing control means is a task. Similarly, with regard to claim 16, an operating system is a software component, while a microprocessor is hardware.

The arguments set forth previously with regard to claim 10 are also applicable to claim 21. In this regard, Applicants further note that the computer provided according to the present invention is, as explained in the specification, used prior to the operating system. That is, the computer is used for initializing the respective nodes which constitute the system, prior to operating the system. Therefore, even after implementation of the system, information in the respective nodes can be changed.

On the other hand, the transmission of the synchronous message by the master node in Green et al. is periodically performed during operation of the system.

Finally, claim 22 recites, in addition to the limitations of claim 21, "a distributed control middleware code generating tool", which receives system configuration information and outputs a program code composed of information to be stored in the module configuration information storing means and the module start control means. Thus, the tool in the present invention is used during implementation of the system, for automatically generating, for example, the module configuration information and program codes for the module start control. Green et al., on the other hand, neither teaches nor suggests any such tools for automatically generating such information and program codes. In particular, Applicants note that the portion of the specification at col. 7, lines 2-24 of Green et al., referred to in the Office Action, relate to data generation by an application program during operation of this system, a matter which is unrelated to the limitations referred to previously in claim 22.

If there are any questions regarding this response or the application in general, a telephone call to the undersigned would be appreciated since this should expedite the prosecution of the application for all concerned.

If necessary to effect a timely response, this paper should be considered as a petition for an Extension of Time sufficient to effect a timely response, and please charge any deficiency in fees or credit any overpayments to Deposit Account No. 05-1323 (Docket # 056207.48110).

Respectfully submitted,

January 18, 2005

Gary R Edwards

Gary R. Edwards
Registration No. 31,824

CROWELL & MORING, LLP
Intellectual Property Group
P.O. Box 14300
Washington, DC 20044-4300
Telephone No.: (202) 624-2500
Facsimile No.: (202) 628-8844
GRE:vgp